

Developing MailManager

Kevin Campbell
Logicalware Ltd

Logicalware

- VC Funded startup company
- One core product - MailManager
 - Open Source Software
 - Runs on linux



MailManager

- Ticketing system for dealing with incoming email
- Incoming mail distributed between end users of the system
- Queue management and filtering allow prioritisation and appropriate assignment of tickets
- Full ticket history available, all activity is journalled
- Standard reply templates available to deal with common queries

The screenshot displays the logicalware mailmanager web interface. At the top, there is a navigation bar with the logo, user and account dropdown menus, and a search box. Below this is a menu with options: TICKETS, SEARCH, REPORTS, HELP, SETTINGS, and LOG OUT. A secondary menu shows filters: OPEN, NEW, OVERDUE, HOLD, CLOSED, ALL, and CREATE. The main content area shows a list of tickets with columns for TICKET, STATUS, SUBJECT, FROM, TIME, DATE, and PRIORITY. The tickets are numbered 001 to 012, with various statuses like OVERDUE, OPEN, and HOLD. An 'OPTIONS' sidebar on the left includes links for 'Change Del', 'Show Head', and 'Messages'.

TICKET	STATUS	SUBJECT	FROM	TIME	DATE	PRIORITY
001	OVERDUE	Problem with Cable Modem	Alex G Bell	19.02.03	16:59	●●●
002	OPEN	Login Problem	Isaac Newton	27.02.03	09:59	●●●
003	OPEN	Urgent technical issue	Andrew Veitch	27.02.03	10:00	●●●
004	OPEN	More information	Andrew Veitch	27.02.03	10:01	●●●
005	CLOSED	Billing question	Adam Smith	27.02.03	10:03	●●●
006	OPEN	Problem listing tickets	Waldemar Hiller	27.02.03	10:04	●●●
007	OPEN	How do I get a full report?	Bob Robinson	27.02.03	10:05	●●●●●
008	OPEN	Password problem now	Isaac Newton	27.02.03	10:06	●●●
009	HOLD	The product's great!	Bill Morrison	27.02.03	10:07	●●
010	OPEN	Help! HELPI	Randy Hopkirk	27.02.03	10:09	●●●
011	OPEN	Issue with incoming email	Andy Flemming	27.02.03	10:29	●●●
012	OPEN	Test message number 1	Andrew Veitch	01.02.03	00:00	●●●

Administration

- Security model restricts access to various parts of the system per user
- Reports information for email throughput and response times
- Reports based on ticket categorisation and service level targets
- Break down of reports by user or account

The screenshot displays the Logicalware MailManager administration interface. The top navigation bar includes 'logicalware mailmanager', 'Users: Show All', and 'Accounts: Show All'. Below this is a 'Get Tickets:' search bar and a menu with 'TICKETS', 'SEARCH', 'REPORTS', 'HELP', 'SETTINGS', and 'LOG OUT'. A secondary menu shows 'USERS | ACCOUNTS | GROUPS | FILTERS | OPTIONS | SYSTEM'. A note states 'Editing permissions are only available to the Administrator'.

The 'OPTIONS' section on the left lists 'List Users', 'Add New User', and 'Remove User'. The main content area features a table with columns: FULL NAME, USER NAME, TICKETS, REPORTS, SETTINGS, and ACTION.

FULL NAME	USER NAME	TICKETS	REPORTS	SETTINGS	ACTION
Administrator	admin	✓	✓	✓	EDIT
Andrew Veitch	andrew	✓	✓	✓	EDIT

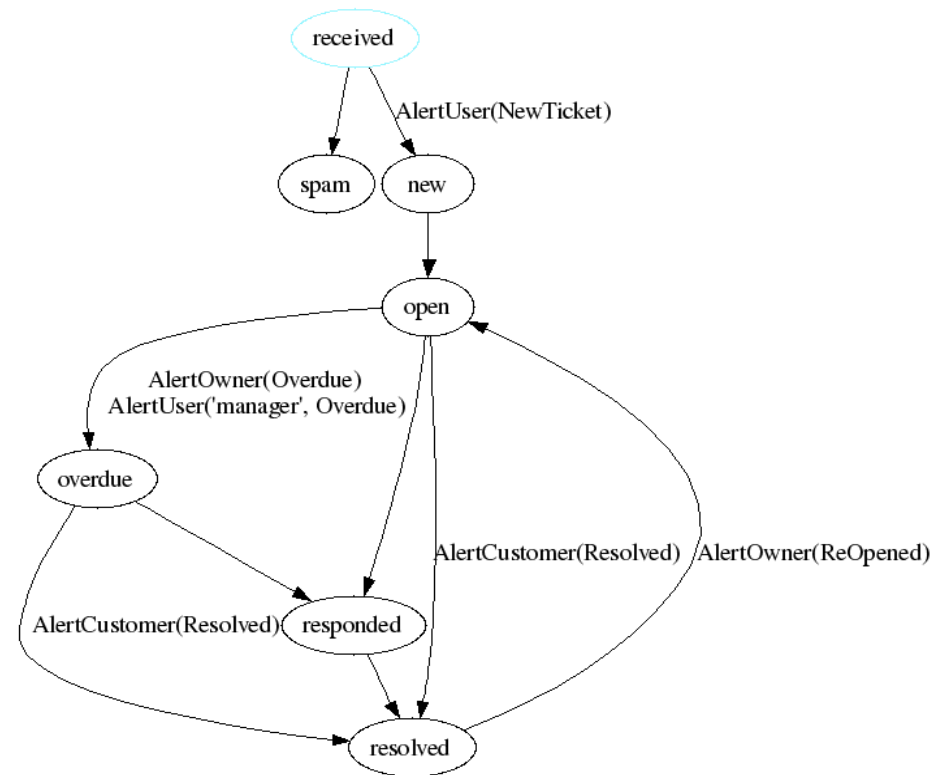
Below the table, the interface shows the user 'andrew' selected, with 'Account: test@logicalprogression.net'. The 'REPORTS' section is active, displaying a bar chart and a list of report options: 'By accounts', 'By users', and 'By category'. The 'PERIOD' section offers 'Daily report' and 'Monthly report'. The 'DATES' section shows 'Date: 01.01.05 to 31.12.05' and 'Select year: 2005'. The 'EXPORT' section includes 'Export to Excel'. The bottom navigation bar contains 'HOME', 'TOP', and 'HELP'.

Main Features

- Platform independent code written in python
- XHTML and CSS based user interface with hotkey support
- Full unicode and i18n support
- Queueing Support for high volume users
- Modular filtering of incoming mail
- Ruleset engine allows extensive customisation
- Supports many backend databases
 - Postgres & MySQL currently supported
 - MS SQL, Oracle & SQLite due soon

Under the hood

- Ruleset engine allows extensible customisation
- Description language allows writing new rulesets without modifying code
- Can be used to integrate with 3rd party systems via RPC interfaces
- Formal description allows more analysis of workflow, additional testing
- Multiple ruleset support due in a later release



Ticket #000452 Lifecycle
2001/01/01 12:00

Company History

- 16th July 2002 – Company Founded
- 16th July 2003 – First Open Source Release
 - *Initial version, ZoDB based product*
- 10th June 2004 - MailManager 1.0 Released
 - *System in production use*
- 12th May 2005 - MailManager 1.1 Released
 - *Added i18n, IMAPS, Notifications*
- 22nd June 2005 – VC Funding received
 - *£300,000 of investment*
 - *Enabled hiring of full time staff*
 -
- 31st Aug 2005 - MailManager 2.0 Released
 - *Major rewrite to work with SQL Backend*
 - *System now dealing with significant volumes of mail*
- 22nd May 2006 - MailManager 2.1 Released
 - *Major restructuring to create modular system*
 - *Added in CSS, Ruleset Engine, Queueing Support*

Business Model

- Hosted Application
 - Web based application, no management involved
- Appliance Servers
 - Black box appliance server provided, which will integrate with existing systems
- Support Contracts
 - Installation of maintenance of software on
- Open Source Users
 - Software released under the GPL
 - Support via bugtracker and mailing lists

Development Challenges

- Hosted Application
 - Direct access/control over all systems
- Appliance Servers
 - Reliance on 3rd party for network access
 - Integration with external systems
- Support Contracts
 - Various target platforms
 - More variable maintenance schedule
- Open Source Users
 - No control over platform at all
 - Testing becomes difficult

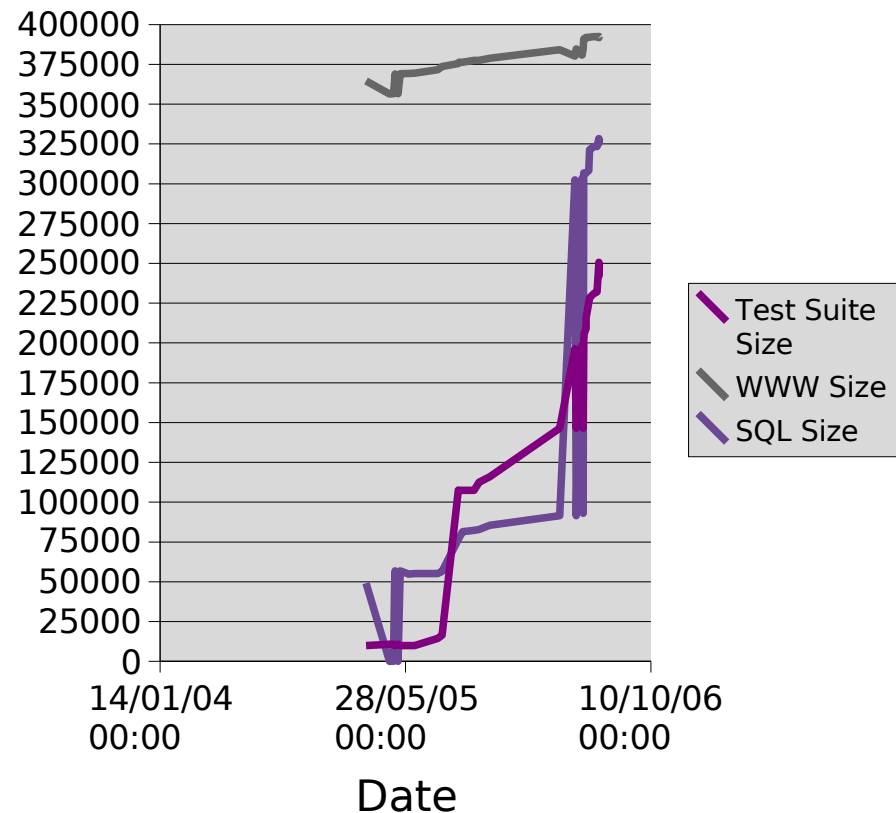
Development Strategy

- Test Driven Development
 - Extensive test code
 - Hook points throughout system
 - Test datasets/known start points
 - Multiple layers of testing
- Continuous Integration
 - Buildbot tracking subversion
 - Large number of platform stacks
 - Virtualisation (VMWare and Xen)

Test Driven Development

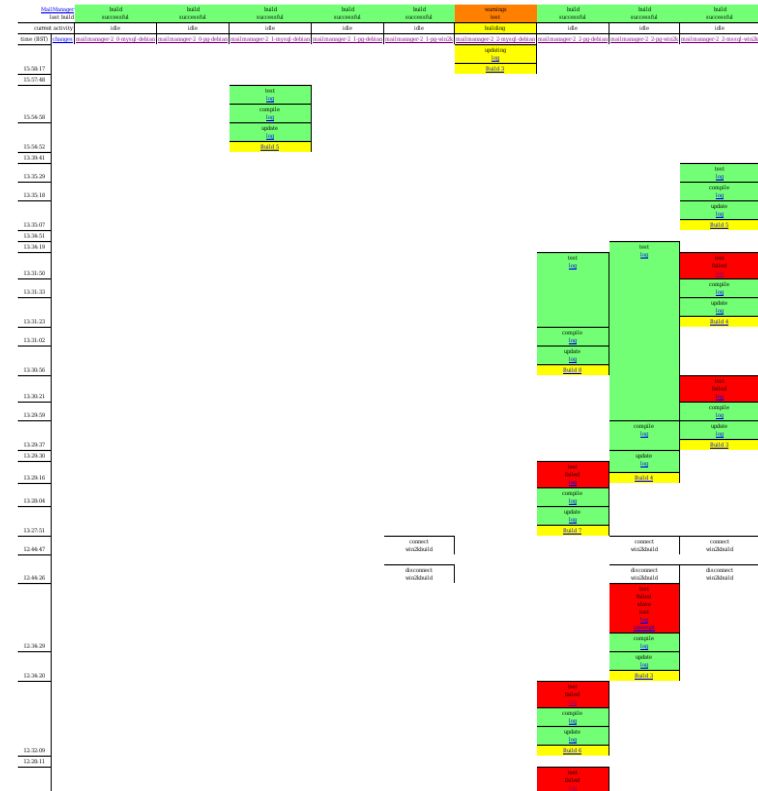
- Original version had almost no test suite code
- Difficult to resolve bugs as setup time for reproduction could be high.
- Developing new features becomes easier as pre-configuration is covered by test suite.
- Prevents regressions of supported functionality
- Test suite currently averages around 10% of the entire code base

Code Base Size



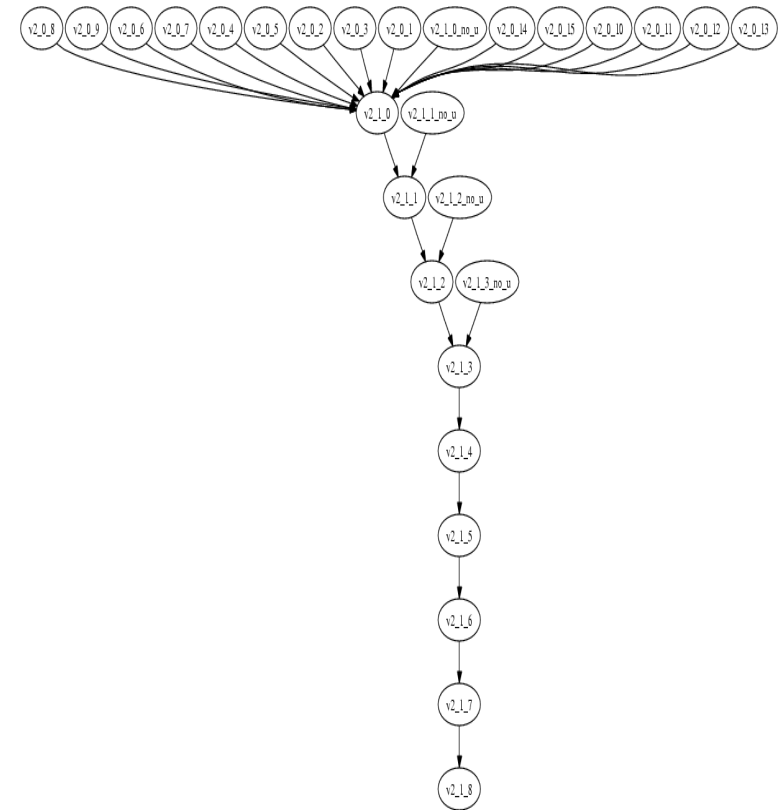
Continuous Integration

- Buildbot runs test suite automatically on source code changes
- Various combinations of
 - MailManager revision
 - Zope version
 - Python version
 - Base operating system
 - SQL database
 - Additional Packages
- Testing done of various platform 'stacks'
- VMWare and Xen used to run multiple operating system revisions
- Email notifications and blame list



Release and Migration Process

- Processes essential for handling multiple revisions in the wild – open source users with many varying versions
- Dataset needs to be maintained between each release.
- Structured Migration system ties into revision control and release management.
- Previously worked with ZoDB, now using structured system based on SQL and python code
- Annotated migration map generated from the code base for administrators.



Migrations

The end

Questions?